

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

**Patent Application**

Appellant(s): Bohannon et al.  
Case: 13-13-1-1  
Serial No.: 10/626,835  
Filing Date: July 24, 2003  
Group: 2161  
Examiner: Paul Kim

Title: Method and Apparatus for Composing XSL Transformations with XML  
Publishing Views

---

TWICE CORRECTED APPEAL BRIEF

Mail Stop Appeal Brief - Patents  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Sir:

In response to the Notice of Non-Compliant Appeal Brief, dated, March 12, 2009, Appellants submit this Corrected Appeal Brief. The Corrected Appeal Brief is substantively identical to the original Appeal Brief, submitted on September 30, 2008, which appealed the final rejection, dated May 7, 2008, of claims 1-18 and 21-32 of the above-identified patent application.

REAL PARTY IN INTEREST

The present application is assigned to Lucent Technologies Inc., as evidenced by an assignment recorded on January 8, 2004 in the United States Patent and Trademark Office at Reel 014870, Frame 0553. The assignee, Lucent Technologies Inc., is the real party in interest.

RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences.

### STATUS OF CLAIMS

The present application was filed on July 24, 2003 with claims 1 through 31. Claims 1-19 and 21-32 are presently pending in the above-identified patent application. Claim 20 was previously cancelled, without prejudice. Claims 1, 10 and 24 were rejected under 35 U.S.C. §103(a) as being unpatentable over Helgeson (United States Patent No. 6,643,652) and in view of Fernandez (United States Patent No. 6,785,673). Claims 2 and 25 were rejected under 35 U.S.C. §103(a) as being unpatentable over Helgeson, in view of Fernandez, and further in view of Chau et al. (United States Publication No. 2002/0123993). Claims 3, 11 and 26 are rejected under 35 U.S.C. §103(a) as being unpatentable over Helgeson, in view of Fernandez and in further view of Jones (United States Publication No. 2004/010754) and further in view of O'Carroll (United States Patent No. 6,772,165). Claims 6, 15 and 29 are rejected under 35 U. S. C. §103(a) as being unpatentable over Helgeson, in view of Fernandez, Jones and of O'Carroll, in further view of Bernstein et al. (United States Patent No. 6,826,568) and further in view of Mani et al. (United States Patent No. 6,654,734). Claims 6, 8, 15, 17, 23 and 29 are rejected under 35 U. S. C. §103(a) as being unpatentable over Fernandez, Jones, O'Carroll, Bernstein and Mani. Claims 9, 18 and 32 were rejected under 35 U.S.C. §103(a) as being unpatentable over Helgeson, in view of Fernandez and in further view of W3C ("XSL Transformation (XSLT), Version 1.0).

Claims 4-5, 7, 12-14, 16, 21-22, 27-28 and 30-31 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims. Claim 19 is allowed. **Claims 8, 17 and 23 appear to be rejected, but they depend on allowed claims 7, 16 and 22, respectively.** Thus, Claims 8, 17 and 23 are also believed to be allowable if rewritten in independent form.

Claims 1, 3, 6, 10-11, 15 and 24, 26, 29 are being appealed.

### STATUS OF AMENDMENTS

The amendments to claims 21 and 22 filed subsequent to the final rejection have been entered. These amendments are believed to have resolved the formal objections.

SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1 requires a method for exporting at least a portion of a relational database (FIG. 2; page 10, lines 7-9, instance, I) to an XML document (FIG. 3, page 7, lines 7-10), comprising the steps of: obtaining an initial view query (FIG. 9, page 11, lines 20-28) that defines an XML view (FIG. 1, page 7, lines 1-4) on said relational database (FIG. 2; page 10, lines 7-9, instance, I) and an XSLT stylesheet specifying at least one transformation (FIG. 5; page 8, lines 5-17); modifying said initial view query (FIG. 9, page 11, lines 20-28) to account for an effect of said at least one transformation (FIG. 5; page 8, lines 5-17); and applying said modified view query to said relational database (FIG. 2; page 10, lines 7-9, instance, I) to obtain said XML document (FIG. 3, page 7, lines 7-10).

Independent claim 10 requires a method for generating a modified view query against a relational database (FIG. 2; page 10, lines 7-9, instance, I) to produce an XML document (FIG. 3, page 7, lines 7-10), comprising the step of: composing an XSLT stylesheet (FIG. 5; page 8, lines 5-17) with an XML view (FIG. 1, page 7, lines 1-4) on said relational database (FIG. 2; page 10, lines 7-9, instance, I) to produce said modified view query.

Independent claim 24 requires a system for exporting at least a portion of a relational database (FIG. 2; page 10, lines 7-9, instance, I) to an XML document (FIG. 3, page 7, lines 7-10), comprising: a memory; and at least one processor, coupled to the memory (page 22, lines 1-22), operative to: obtain an initial view query (FIG. 9, page 11, lines 20-28) that defines an XML view (FIG. 1, page 7, lines 1-4) on said relational database (FIG. 2; page 10, lines 7-9, instance, I) and an XSLT stylesheet specifying at least one transformation (FIG. 5; page 8, lines 5-17); modify said initial view query (FIG. 9, page 11, lines 20-28) to account for an effect of said at least one transformation (FIG. 5; page 8, lines 5-17); and apply said modified view query to said relational database (FIG. 2; page 10, lines 7-9, instance, I) to obtain said XML document (FIG. 3, page 7, lines 7-10).

Dependent claim 3 requires the steps of generating a first graph representing processing done by said XSLT stylesheet; and combining said first graph with a second graph representing said initial view query by matching pairs of nodes from the first and second graphs (FIG. 11, page 6, lines 1-5).

Dependent claim 6 requires the steps of pruning said combined graph to remove unnecessary nodes (page 6, line 5); and modifying said combined graph to produce a modified view query that handles formatting instructions (page 6, line 6).

Dependent claim 11 requires the steps of generating a first graph representing processing done by said XSLT stylesheet; and combining said first graph with a second graph representing an initial view query that defines said XML view on said relational database by matching pairs of nodes from the first and second graphs (FIG. 11, page 6, lines 1-5).

Dependent claim 15 requires the steps of pruning said combined graph to remove unnecessary nodes (page 6, line 5); and modifying said combined graph to produce a modified view query that handles formatting instructions (page 6, line 6).

Dependent claim 26 requires that the processor is further operative to generate a first graph representing processing done by said XSLT stylesheet; and combine said first graph with a second graph representing said initial view query by matching pairs of nodes from the first and second graphs (FIG. 11, page 6, lines 1-5).

Dependent claim 29 requires that the processor is further operative to prune said combined graph to remove unnecessary nodes (page 6, line 5); and modify said combined graph to produce a modified view query that handles formatting instructions (page 6, line 6).

#### STATEMENT OF GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1, 10 and 24 were rejected under 35 U.S.C. §103(a) as being unpatentable over Helgeson and in view of Fernandez. Claims 3, 11 and 26 are rejected under 35 U.S.C. §103(a) as being unpatentable over Helgeson, in view of Fernandez and in further view of Jones and further in view of O'Carroll. Claims 6, 15 and 29 are rejected under 35 U. S. C. §103(a) as being unpatentable over Helgeson, in view of Fernandez, Jones and of O'Carroll, in further view of Bernstein et al. and further in view of Mani et al.

#### ARGUMENT

##### Independent Claims

Independent claims 1, 10 and 24 were rejected under 35 U.S.C. §103(a) as being anticipated by Helgeson in view of Fernandez.

*Claims 1 and 24*

With regard to claim 1, for example, the Examiner asserts that Helgeson teaches a method for exporting at least a portion of a relational database to an XML document. Helgeson is directed to methods and apparatus for managing data exchange among systems in a network.

With regards to claims 1 and 24, the Examiner asserts that Fernandez, column 2, lines 65-67, and col. 3, lines 23-54, disclose an “initial view query” that defines an XML view on a relational database. The Examiner further asserts that Helgeson, columns 49 and 73-74 disclose “modifying the initial view query to account for an effect of said at least one transformation (specified in an XLST stylesheet).”

As indicated in Appellants’ prior responses, a “view query” specifies a *mapping between the relational tables and the resulting XML document*. Helgeson does not disclose or suggest view queries. Column 80, lines 51-55, of Helgeson merely discloses that an XML document can be created from a database. While Helgeson may use the term “view” in the presentation sense, Helgeson does not use the term “view query,” nor does Helgeson address using view queries to map between relational tables and a resulting XML document.

In the Advisory Action, the Examiner again asserts that Appellants are arguing features not recited in the claims (referencing the mapping between tables and an XML document). To the contrary, however, Appellants are merely referencing a well-accepted definition of the term “view query” to those of ordinary skill in the art. The Examiner has not contested this well-accepted definition of the term “view query.”

In addition, since Helgeson is not addressing “view queries,” Helgeson does not disclose or suggest “***modifying the initial view query*** to account for an effect of said at least one transformation (specified in an XLST stylesheet),” or “applying said modified view query to said relational database to obtain said XML document,” as further required by independent claims 1 and 24.

Likewise, Appellants can find no disclosure (and the Examiner has not alleged any) in Fernandez of “***modifying the initial view query*** to account for an effect of said at least one transformation (specified in an XLST stylesheet),” or “applying said modified view query to said relational database to obtain said XML document,” as further required by independent claims 1 and 24.

Fernandez merely teaches that an XML view can be specified by a query in a declarative query language of a middleware system. See, col. 3, lines 25-27. There is no disclosure or suggestion

that this view query is modified at all, or especially, “to account for an effect of said at least one transformation.”

In the Response to Arguments section, the Examiner references col. 3, lines 11-22 of Fernandez for its disclosure of mapping relational sources to XML views. Appellants can find no teaching in this passage, however, of modifying a *view query*.

Thus, even as combined in the manner suggested by the Examiner Helgeson and Fernandez *do not teach every element of the independent claims*. Furthermore, based on the KSR considerations discussed hereinafter, the combination/modification suggested by the Examiner is not appropriate. Other than to allege that the motivation to combine is the desire to obtain the result (“in order to obtain an XML document according to the transformation specified by an XSLT stylesheet”), the Examiner has failed to establish “an apparent reason to combine ... known elements.” *KSR International Co. v. Teleflex Inc. (KSR)*, 550 U.S. \_\_\_, 82 USPQ2d 1385 (2007). Appellants query how this suggests an alleged combination to *modify the initial view query* to account for an effect of said at least one transformation (specified in an XSLT stylesheet),” or “applying said modified view query to said relational database to obtain said XML document.” As discussed below, this is insufficient to satisfy the Examiner’s burden of proof under *KSR*.

Appellants are claiming a new technique for exporting at least a portion of a relational database to an XML document **by** “*modifying the initial view query* to account for an effect of said at least one transformation (specified in an XSLT stylesheet),” and “applying said modified view query to said relational database **to obtain** said XML document.”

Initial view queries are not merely recited as an element of each claim, but rather the functional language requires that the initial view queries be modified to account for a transformation and then applied to a relational database to obtain the result. There is no suggestion in Helgeson or in Fernandez, alone or in combination, to modify initial view queries to account for a transformation and then applied to a relational database to obtain the result. In addition, there is absolutely no suggestion to modify the *views* (purely in the presentation sense) of Helgeson with the *initial view queries* of Fernandez.

Finally, the Examiner asserts that the limitation “to obtain said XML document” should not be given patentable weight, as it is “an intended use.” To the contrary, however, this limitation

provides additional functional language to describe the present invention and should be given patentable weight.

*Claim 10*

With regards to claim 10, the Examiner again asserts that Helgeson discloses composing an XSLT stylesheet (citing col. 51, lines 32-34) and Fernandez discloses (with) an XML view on said relational database to produce said modified view query (citing column 2, lines 65-67, and col. 3, lines 23-54). Appellants can find no disclosure or suggestion in Fernandez of generating a modified view query. The passages referenced by the Examiner merely teach that an XML view can be specified by a query in a declarative query language of a middleware system. See, col. 3, lines 25-27. There is no disclosure or suggestion that this view query is **modified**. Claim 10 emphasizes that the modified view query is generated against a relational database to produce an XML document.

Appellants submit that the KSR considerations addressed above with regard to claims 1 and 24 are equally applicable to claim 10.

Appellants respectfully request the withdrawal of the rejection of independent claims 1, 10, and 24.

Dependent Claims

Claims 2-9, 11-18, 20-23 and 25-32 are dependent on independent claims 1, 10, 19 and 24, respectively, and are therefore patentably distinguished over each of the cited references, alone or in combination, because of their dependency from independent claims 1, 10, 19 and 24, for the reasons set forth above, as well as other elements these claims add in combination to their base claim.

Claims 4-5, 7, 12-14, 16, 21-22, 27-28 and 30 were indicated to be allowable if rewritten in independent form. **Claims 8, 17 and 23 appear to be rejected, but they depend on allowed claims 7, 16 and 22, respectively.** Thus, Claims 8, 17 and 23 are also believed to be allowable if rewritten in independent form.

Dependent claims 3, 11 and 26 require the steps of generating a first graph representing processing done by said XSLT stylesheet; and combining said first graph with a second graph representing said initial view query by matching pairs of nodes from the first and second graphs. Contrary to the Examiner's assertion, O'Carroll does not disclose or suggest an initial view query at all, nor a second graph representing said initial view query. A "view query" specifies a *mapping between the relational tables and the resulting XML document*. This is not shown by O'Carroll.

Dependent claims 6, 15 and 29 require the steps of pruning said combined graph to remove unnecessary nodes; and modifying said combined graph to produce a modified view query that handles formatting instructions. Contrary to the Examiner's assertion, Mani does not disclose or suggest a modified view query at all, nor a modified view query that handles formatting instructions. A "view query" specifies a *mapping between the relational tables and the resulting XML document*. This is not shown by Mani.

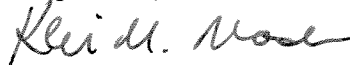
Conclusion

All of the pending claims following entry of the amendments, i.e., claims 1-19 and 21-32, are in condition for allowance and such favorable action is earnestly solicited.

If any outstanding issues remain, or if the Examiner or the Appeal Board has any further suggestions for expediting allowance of this application, the Examiner and the Appeal Board are invited to contact the undersigned at the telephone number indicated below.

The attention of the Examiner and the Appeal Board to this matter is appreciated.

Respectfully,



Kevin M. Mason  
Attorney for Appellant(s)  
Reg. No. 36,597  
Ryan, Mason & Lewis, LLP  
1300 Post Road, Suite 205  
Fairfield, CT 06824  
(203) 255-6560

Date: April 13, 2009



APPENDIX

1. A method for exporting at least a portion of a relational database to an XML document, comprising the steps of:
  - obtaining an initial view query that defines an XML view on said relational database and an XSLT stylesheet specifying at least one transformation;
  - modifying said initial view query to account for an effect of said at least one transformation; and
  - applying said modified view query to said relational database to obtain said XML document.
2. (Original) The method of claim 1, wherein said XSLT stylesheet is based on a restrictive subset of XSLT.
3. The method of claim 1 further comprising the steps of generating a first graph representing processing done by said XSLT stylesheet; and combining said first graph with a second graph representing said initial view query by matching pairs of nodes from the first and second graphs.
4. The method of claim 3, wherein said combined graph is a context transition graph for an XSLT stylesheet executed on said initial view query.
5. The method of claim 4, wherein said context transition graph captures context transitions that occur when evaluating said XSLT stylesheet on said XML document produced by said initial view query.
6. The method of claim 3, further comprising the steps of pruning said combined graph to remove unnecessary nodes; and modifying said combined graph to produce a modified view query that handles formatting instructions.
7. The method of claim 6, further comprising the step of generating a traverse view query from a context transition graph prior to generating said modified view query, said traverse view query

capturing traversal actions of said XSLT stylesheet on said XML document produced by said initial view query.

8. The method of claim 7, wherein said formatting instructions are expressed as output tag trees for each node in said traverse view query; and further comprising the step of combining said output tag trees and said traverse view query to generate said modified view query.

9. The method of claim 1, wherein said obtained XML document is substantially similar to a second XML document produced by applying said XSLT stylesheet on said XML document produced by said initial view query.

10. A method for generating a modified view query against a relational database to produce an XML document, comprising the step of:

composing an XSLT stylesheet with an XML view on said relational database to produce said modified view query.

11. The method of claim 10, further comprising the steps of generating a first graph representing processing done by said XSLT stylesheet; and combining said first graph with a second graph representing an initial view query that defines said XML view on said relational database by matching pairs of nodes from the first and second graphs.

12. The method of claim 11, wherein said combined graph is a context transition graph for an XSLT stylesheet executed on said initial view query.

13. The method of claim 12, wherein said context transition graph captures context transitions that occur when evaluating said XSLT stylesheet on said XML document produced by said initial view query.

14. The method of claim 11, wherein said context transition graph includes selecting and matching transformations from said XSLT stylesheet.

15. The method of claim 11, further comprising the steps of pruning said combined graph to remove unnecessary nodes; and modifying said combined graph to produce a modified view query that handles formatting instructions.

16. The method of claim 15, further comprising the step of generating a traverse view query from a context transition graph prior to generating said modified view query, said traverse view query capturing traversal actions of said XSLT stylesheet on said XML document produced by said initial view query.

17. The method of claim 16, wherein said formatting instructions are expressed as output tag trees for each node in said traverse view query; and further comprising the step of combining said output tag trees and said traverse view query to generate said modified view query.

18. The method of claim 10, wherein an obtained XML document is substantially similar to a second XML document produced by applying said XSLT stylesheet on said XML document produced by said initial view query.

19. A method for generating a modified view query against a relational database to produce an XML document, comprising the steps of:

generating a first graph representing processing done by an XSLT stylesheet;

combining said first graph with a second graph to generate a combined graph representing a view query that defines an XML view on said relational database by matching pairs of nodes from the first and second graphs, wherein said combined graph is a context transition graph for said XSLT stylesheet executed on an initial view query that defines said XML view on said relational database;

pruning said combined graph to remove unnecessary nodes; and

modifying said combined graph to produce said modified view query that handles formatting instructions.

20. (Cancelled).

21. The method of claim 19, wherein said context transition graph captures context transitions that occur when evaluating said XSLT stylesheet on said XML document produced by said initial view query.

22. The method of claim 19, further comprising the step of generating a traverse view query from said context transition graph prior to generating said modified view query, said traverse view query capturing traversal actions of said XSLT stylesheet on said XML document produced by said initial view query.

23. The method of claim 22, wherein said formatting instructions are expressed as output tag trees for each node in said traverse view query; and further comprising the step of combining said output tag trees and said traverse view query to generate said modified view query.

24. A system for exporting at least a portion of a relational database to an XML document, comprising:

a memory; and

at least one processor, coupled to the memory, operative to:

obtain an initial view query that defines an XML view on said relational database and an XSLT stylesheet specifying at least one transformation;

modify said initial view query to account for an effect of said at least one transformation;

and

apply said modified view query to said relational database to obtain said XML document.

25. The system of claim 24, wherein said XSLT stylesheet is based on a restrictive subset of XSLT.

26. The system of claim 24, wherein said processor is further operative to generate a first graph representing processing done by said XSLT stylesheet; and combine said first graph with a second graph representing said initial view query by matching pairs of nodes from the first and second graphs.

27. The system of claim 26, wherein said combined graph is a context transition graph for an XSLT stylesheet executed on said initial view query.
28. The system of claim 27, wherein said context transition graph captures context transitions that occur when evaluating said XSLT stylesheet on said XML document produced by said initial view query.
29. The system of claim 26, wherein said processor is further operative to prune said combined graph to remove unnecessary nodes; and modify said combined graph to produce a modified view query that handles formatting instructions.
30. The system of claim 29, wherein said processor is further configured to generate a traverse view query from a context transition graph prior to generating said modified view query, said traverse view query capturing traversal actions of said XSLT stylesheet on said XML document produced by said initial view query.
31. The system of claim 30, wherein said formatting instructions are expressed as output tag trees for each node in said traverse view query; and further comprising the step of combining said output tag trees and said traverse view query to generate said modified view query.
32. The system of claim 24, wherein said obtained XML document is substantially similar to a second XML document produced by applying said XSLT stylesheet on said XML document produced by said initial view query.

EVIDENCE APPENDIX

There is no evidence submitted pursuant to § 1.130, 1.131, or 1.132 or entered by the Examiner and relied upon by appellant.

RELATED PROCEEDINGS APPENDIX

There are no known decisions rendered by a court or the Board in any proceeding identified pursuant to paragraph (c)(1)(ii) of 37 CFR 41.37.